PATENT OFFICE

JAPANESE GOVERNMENT

This is to certify that the annexed is a true copy of the following application as filed with this Office.

Date of Application: September 30, 1999

Application Number: Patent Application No. 11-279208

Applicant(s): CASIO COMPUTER CO., LTD.

April 7, 2000

Commissioner,

Patent Office Takahiko Kondo

Certificate No. 2000-3024156

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。
This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: １９９９年 ９月３０日

出願番号
Application Number: 平成１１年特許願第２７９２０８号

出願人
Applicant (s): カシオ計算機株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

２０００年 ４月 ７日

特許庁長官
Commissioner,
Patent Office 近藤隆彦

出証番号 出証特２０００－３０２４１５６

【書類名】　特許願

【整理番号】　A009904569

【提出日】　平成11年 9月30日

【あて先】　特許庁長官　殿

【国際特許分類】　G06F 12/00

G09C 1/00

【発明の名称】　データベース管理装置、データベースシステム、暗号化装置及び記録媒体

【請求項の数】　9

【発明者】

【住所又は居所】　東京都羽村市栄町３丁目２番１号　カシオ計算機株式会社羽村技術センター内

【氏名】　佐藤　誠

【発明者】

【住所又は居所】　東京都羽村市栄町３丁目２番１号　カシオ計算機株式会社羽村技術センター内

【氏名】　竹田　恒治

【特許出願人】

【識別番号】　000001443

【氏名又は名称】　カシオ計算機株式会社

【代理人】

【識別番号】　100058479

【弁理士】

【氏名又は名称】　鈴江　武彦

【電話番号】　03-3502-3181

【選任した代理人】

【識別番号】　100084618

【弁理士】

【氏名又は名称】　村松　貞男

【選任した代理人】

【識別番号】　100068814

【弁理士】

【氏名又は名称】　坪井　淳

【選任した代理人】

【識別番号】　100092196

【弁理士】

【氏名又は名称】　橋本　良郎

【選任した代理人】

【識別番号】　100091351

【弁理士】

【氏名又は名称】　河野　哲

【選任した代理人】

【識別番号】　100088683

【弁理士】

【氏名又は名称】　中村　誠

【手数料の表示】

【予納台帳番号】　011567

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9005919

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　データベース管理装置、データベースシステム、暗号化装置及び記録媒体

【特許請求の範囲】

【請求項１】　データベースの所定の列項目のデータについて当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては各行毎に固有の行鍵を用いて暗号化する暗号化手段と、

この暗号化手段によって暗号化されたデータベースを記憶する記憶手段と

を具備したことを特徴とするデータベース管理装置。

【請求項２】　前記所定の列項目を検索対象とした場合に、検索用として入力されたデータを前記所定の列項目に共通の列鍵を用いて暗号化し、その暗号化された検索用データと前記記憶手段に記憶された暗号化データベースの各項目データとを比較して検索処理を行うデータベース検索手段を具備したことを特徴とする請求項１記載のデータベース管理装置。

【請求項３】　前記暗号化手段は、データベースの所定の列項目のデータを当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては各行毎に固有の行鍵と当該列項目に共通の列鍵とを複合的に用いて暗号化することを特徴とする請求項１記載のデータベース管理装置。

【請求項４】　前記暗号化手段は、所定の関数に基づいて多次元空間に逐次ベクトルを発生させ、これらのベクトル成分を暗号のキーストリームとした暗号化方式方式を用い、前記行鍵および列鍵を前記関数の定数としてデータベースの暗号化を行うことを特徴とする請求項１記載のデータベース管理装置。

【請求項５】　データベースを有する第１の情報端末と、この第１の情報端末にデータベースの検索を依頼する第２の端末とを有し、これらをネットワークを介して接続してなるデータベースシステムにおいて、

前記第１の情報端末側で、前記データベースの所定の列項目のデータについて当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては各行毎に固有の行鍵を用いて暗号化し、

前記第２の情報端末から前記所定の列項目を検索対象としてデータベースの検

索を依頼する際に、検索用として入力されたデータを当該列項目に共通の列鍵を用いて暗号化し、その暗号化された検索用データを前記ネットワークを介して前記第１の情報端末に送り、

前記第１の情報端末側で、前記検索用データに基づいて前記暗号化データベースの検索処理を行い、その検索結果として得られたデータを暗号化された状態で前記ネットワークを介して前記第２の情報端末に返信することを特徴とするデータベースシステム。

【請求項６】　所定の列項目のデータが当該列項目に共通の列鍵を用いて暗号化されているデータベースを管理するデータベース管理装置であって、

所定の列項目を対象にしたデータ検索をする際、検索用に入力されたデータを前記列鍵を用いて暗号化する暗号化手段と、

この暗号化された検索用データと前記暗号化されているデータベースの各項目データとを比較して検索処理を行う検索手段と

を具備したことを特徴とするデータベース管理装置。

【請求項７】　データベースの少なくとも１つの列項目データを当該列項目に共通の列鍵を用いて暗号化する暗号化装置であって、

暗号化の対象となる原データを取得する原データ取得手段と、

少なくとも前記列鍵によってを決定される関数を用いて、ｎ（ｎ≧１）次元の空間の閉領域内に定義されたベクトルを逐次的に生成するベクトル生成手段と、

前記原データ取得手段によって得られた原データと前記ベクトル生成手段によって生成されたベクトルの成分との論理演算をビット単位で行って暗号データを生成する論理演算手段と

を具備したことを特徴とする暗号化装置。

【請求項８】　コンピュータに、

データベースの所定の列項目のデータについて当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては各行毎に固有の行鍵を用いて暗号化する暗号化機能と、

この暗号化機能の実行結果として得られた暗号化データベースに対するデータ検索を行う検索機能と

を実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項９】　所定の列項目のデータが当該列項目に共通の列鍵を用いて暗号化されているデータベースを管理するためのコンピュータに、

所定の列項目を対象にしたデータ検索をする際、検索用に入力されたデータを前記列鍵を用いて暗号化する暗号化機能と、

この暗号化された検索用データと前記暗号化されているデータベースの各項目データとを比較して検索処理を行う検索機能と

を実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、データベースを暗号化して管理するデータベース管理装置と、暗号化データベースを有する検索装置をネットワーク上に配置したデータベースシステムと、データベースの暗号化を行う暗号化装置及び記録媒体に関する。

【０００２】

【従来の技術】

データベース管理装置では、データベースのセキュリティを確保するため、管理対象として保有しているデータベースを暗号化して保存しておくことが求められる。

【０００３】

ここでセキュリティを高めるためには、より複雑な暗号処理を施すことで実現することができるが、複雑な暗号処理は演算に要する時間も大きくなることは避けられない。

【０００４】

データベースは膨大なデータを有し、データ検索は、それらのデータの中からある項目について条件に合致するものを選び出し、条件に合う項目データを含むレコード（行データ）を出力するという処理になる。従って、大量のデータを取

り扱うデータ検索システムにおいては演算時間の増加は処理効率を悪化させる原因となる。

【０００５】

【発明が解決しようとする課題】

上述したように、機密性の高いデータを記憶しているデータベースにはセキュリティの要求があり、セキュリティを高めるための暗号化処理はデータベースの利用性を妨げるという問題がある。

【０００６】

本発明は前記のような課題を解決するためになされたもので、データベースのセキュリティを確保すると共に、データの高速検索を可能にしたデータベース管理装置、データベースシステム、暗号化装置及び記録媒体を提供することを目的とする。

【０００７】

【課題を解決するための手段】

本発明は、データベースを暗号化する際に、検索に利用される列項目のデータについては当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては各行毎に固有の行鍵を用いて暗号化するようにしたものである。

【０００８】

具体的には、本発明のデータベース管理装置は、データベースの所定の列項目のデータについて当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては各行毎に固有の行鍵を用いて暗号化する暗号化手段と、この暗号化手段によって暗号化されたデータベースを記憶する記憶手段とで構成される。

【０００９】

このような構成されよれば、データベースを暗号化する際に、各行毎に鍵を異ならせることでセキュリティを高めることができ、検索時には、検索用として入力されたデータを前記所定の列項目に共通の列鍵を用いて暗号化し、その暗号化された検索用データと暗号化データベースの各項目データとを比較することで高速検索を実現することができる。

【００１０】

また、検索に利用される列項目以外の列項目のデータについて、各行毎に固有の行鍵と当該列項目に共通の列鍵とを複合的に用いて暗号化するようにすれば、さらにセキュリティを強化することができる。

【００１１】

また、暗号化方式として、所定の関数に基づいて多次元空間に逐次ベクトルを発生させ、これらのベクトル成分を暗号のキーストリームとした多次元空間回転方式を用いれば、複雑な演算処理を必要とせずにデータベースを暗号化することができるので、演算能力の低い情報処理装置にも適用可能となる。

【００１２】

また、データベースの保管場所を離間させておき、別の情報端末からネットワークを介して検索を依頼するようなデータベースシステムを構築することも可能である。この場合、前記のようにデータベースの所定の列項目（検索に利用される列項目）のデータを当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては各行毎に固有の行鍵を暗号化しておく。そして、別の情報端末からデータベースの検索を依頼する際に、検索用データを当該列項目に共通の列鍵を用いて暗号化し、その暗号化された検索用データをネットワークを介して送るようにする。この検索用データを受けることで、前記暗号化データベースの検索処理を行い、その検索結果として得られたデータを暗号化された状態でネットワークを介して前記情報端末に返信する。したがって、常に暗号化された状態でデータをやり取りできるので、データベースのセキュリティを確保することができる。

【００１３】

【発明の実施の形態】

以下、図面を参照して本発明の実施形態を説明する。

【００１４】

（第１の実施形態）

図１は本発明の第１の実施形態に係るデータベース管理装置の構成を示す図である。本装置は、行と列からなるマトリクス形式のデータベースを有し、そのデ

ータベースを暗号化して管理する機能と共に、入力された検索用データを暗号化して、その暗号化された検索用データに基づいてデータベースを検索する機能を備えたものであって、例えば磁気ディスク等の記録媒体に記録されたプログラムを読み込み、このプログラムによって動作が制御されるコンピュータによって実現される。

【００１５】

図１に示すように、本装置には、ＣＰＵ１１、表示装置１２、入力装置１３、プログラム記憶装置１４、鍵記憶装置１５、データ記憶装置１６が設けられている。

【００１６】

ＣＰＵ１１は、本装置全体の制御を行うものであり、プログラム記憶装置１４に記憶されたプログラムを読み込み、そのプログラムに従って各種処理を実行する。本実施形態において、ＣＰＵ１１は図２に示すようなデータベースの暗号化処理や、図３および図４に示すようなデータベースの検索処理を実行する。

【００１７】

表示装置１２は、データを表示するためのデバイスであり、例えばＬＣＤ（Liquid Crystal Display）やＣＲＴ（Cathode-ray tube）等が用いられる。入力装置１３は、データを入力するためのデバイスであり、例えばキーボード、マウス等が用いられる。

【００１８】

プログラム記憶装置１４は、例えばＲＯＭあるいはＲＡＭなどで構成され、本装置に必要なプログラムを記憶する。本装置に必要なプログラムとしては、データベース管理プログラムや暗号化プログラム等がある。

【００１９】

なお、プログラム記憶装置１４は半導体メモリの他に磁気的、光学的記録媒体で構成することができる。この記録媒体はＣＤ-ＲＯＭ等の可搬型の媒体やハードディスク等の固定的な媒体を含む。また、この記録媒体に格納するプログラムは、その一部若しくは全部をサーバやクライアントからネットワーク回線などの伝送媒体を介して伝送制御部から受信する構成にしてもよく、更に、前記記録媒

体はネットワーク上に構築されたサーバの記録媒体であってもよい。更に、前記プログラムをネットワーク回線などの伝送媒体を介してサーバーやクライアントへ伝送してこれらの機器にインストールするように構成してもよい。

【００２０】

鍵記憶装置１５は、例えばＲＡＭなどで構成され、データベースの暗号化に用いられる鍵（行鍵と列鍵）を記憶する。

【００２１】

データ記憶装置１６は、本装置に必要な各種のデータを記憶するためのデバイスであり、例えばＲＡＭあるいは磁気ディスク装置等の外部記憶装置で構成される。このデータ記憶装置１６には、データベースを格納しておくためのデータベース格納エリア１６ａ、データベースを暗号化する際にオペレータにより設定された情報（検索対象項目、非暗号化項目等）を格納しておくための暗号化設定情報格納エリア１６ｂ、データベースを検索する際にオペレータにより設定された情報（対象列項目、検索文字列等）を格納しておくための検索設定情報格納エリア１６ｃ、データベース検索時の比較文字列を格納しておくための比較文字列格納エリア１６ｄなどが設けられている。

【００２２】

ここで、本装置の動作を説明する前に、本装置によって実現されるデータベースの暗号化方法について説明しておく。

【００２３】

データベースを暗号化する場合に、各行毎（レコード毎）に異なる鍵を用いて暗号化すると、鍵の解読は困難となり、セキュリティ性を高めることができる。しかし、データベースを検索する際に、暗号化されているデータを各行毎の鍵で復号化するか、または、検索用として入力されたデータ（キーワード）を各行毎の鍵で暗号化しなければならいため、検索結果を得るのに時間がかかることになる。一方、各列毎に異なる鍵を用いてデータベースを暗号化すると、検索対象となる列項目に対応した鍵のみを用いて検索用データを暗号化すれば良いのでデータベース検索を高速に行うことができる。しかし、同じ列の中で同一のデータがあると、暗号化結果も同じになってしまうため、そこから鍵を解読される可能性

が高くなる。

【００２４】

そこで、本発明では、データベースを暗号化する際に、検索に頻繁に利用される列項目のデータについては列共通鍵で暗号化し、その他の列項目のデータについては各行毎に異なる鍵を与えて暗号化することを特徴とする。つまり、各行毎に鍵を異ならせることで、セキュリティを高めると共に、検索時には検索に利用される項目に入力されたデータを列鍵で暗号化してデータベース上の暗号化データと比較することで高速検索を可能とするものである。

【００２５】

図５は第１の実施形態におけるデータベースの構成を説明するための図であり、図５（ａ）は暗号化前の状態、同図（ｂ）は暗号化後の状態、同図（ｃ）は復号化後の状態を示している。また、図６は第１の実施形態における列鍵と行鍵の構成を示す図である。

【００２６】

図５（ａ）に示すように、本装置では、行と列からなるマトリクス形式のデータベースを備えている。ここでは、個人データをデータベース化した場合を示している。このデータベースは、１レコードが「ｎｕｍｂｅｒ」，「ｎａｍｅ」，「ｓｔａｔｅ」，「ｗｅｉｇｈｔ」，「ｈｅｉｇｈｔ」，「ａｇｅ」，「ｐｈｏｎｅ」の各項目で構成するようにしている。

【００２７】

このようなデータベースに対し、列鍵と行鍵を用いて暗号化を行う。すなわち、検索に頻繁に利用される列項目を「ｎａｍｅ」，「ｓｔａｔｅ」，「ａｇｅ」とした場合に、これらの列項目の各行のデータについては、図６に示すように、「ａｐｐｌｅ」，「ｏｒａｎｇｅ」，「ｌｅｍｏｎ」といった各列項目に共通の列鍵を用いて暗号化し、その他の列項目「ｗｅｉｇｈｔ」，「ｈｅｉｇｈｔ」，「ｐｈｏｎｅ」の各行のデータについては、各行毎に固有の鍵を用いて暗号化する。

【００２８】

なお、「ｎｕｍｂｅｒ」の列は暗号化を行わないものとする。行鍵としては、

「ｔｉｇｅｒ」，「ｄｏｇ」，「ｃａｔ」，「ｍｏｕｓｅ」，「ｅｌｅｐｈａｎｔ」，「ｃｏｗ」，「ｐｉｇ」，「ｒａｂｂｉｔ」，「ｌｉｏｎ」といった鍵が用いられる。これらの鍵は所定の関数を用いて各列または各行毎に数学的に発生させることができる。

【００２９】

図５（ａ）のデータベースを列鍵と行鍵を用いて暗号化した結果を図５（ｂ）に示す。データ記憶装置１６のデータベース格納エリア１６ａには、図５（ｂ）に示すような状態でデータベースが保存されることになる。

【００３０】

データベースを検索する場合には、検索に利用される列項目に対応した列鍵を用いて検索用データを暗号化してから検索処理を行う。例えば、「ｓｔａｔｅ」の中の「Ｆｌｏｒｉｄａ」といったデータを検索する場合には、まず、検索用データとして入力された「Ｆｌｏｒｉｄａ」を「ｓｔａｔｅ」の列鍵「ａｐｐｌｅ」で暗号化して、「ｈ＊／ｆＤＤ」を得る。この「ｈ＊／ｆＤＤ」といったデータを「ｓｔａｔｅ」の列の各行から検索する。これにより、「ｎｕｍｂｅｒ２」と「ｎｕｍｂｅｒ８」に該当するデータが存在することがわかる。

【００３１】

また、暗号化されたデータベースを元に戻す場合には、暗号化の時と同じ列鍵と行鍵を用いる。図５（ｂ）のデータベースを暗号化の時と同じ列鍵と行鍵を用いて復号化すると、図５（ｃ）に示すように元のデータを得ることができる。

【００３２】

以下に、本装置の動作について説明する。

【００３３】

ここでは、（ａ）データベースを暗号化する場合の処理と、（ｂ）データベースを検索する場合の処理に分けて説明する。なお、このフローチャートで示す各機能を実現するプログラムはＣＰＵが読み取り可能なプログラムコードの形態で前記プログラム記憶装置１４の記録媒体に格納されている。また、このプログラムはプログラムコードの形態でネットワーク回線などの伝送媒体を介して伝送することもできる。

【００３４】

（ａ）データベースを暗号化する場合

図２は本装置にて実行されるデータベース暗号化処理の動作を示すフローチャートである。今、データベースが暗号化されていない状態でデータ記憶装置１６のデータベース格納エリア１６ａに記憶されているものとする。この状態が図５（ａ）である。

【００３５】

まず、図示せぬデータベース暗号化設定画面にて、暗号化対象となるデータベースを指定する（ステップＡ１１）。

【００３６】

次に、そのデータベースに設けられた各列項目の中で検索に利用する列項目と、暗号化を必要としない列項目をそれぞれ設定する（ステップＡ１２）。図５（ａ）の例では、検索に利用する列項目は「ｎａｍｅ」，「ｓｔａｔｅ」，「ａｇｅ」であり、暗号化を必要としない列項目は「ｎｕｍｂｅｒ」である。ここでの設定情報はデータ記憶装置１６の暗号化設定情報格納エリア１６ｂに記憶される。

【００３７】

次に、当該データベースの暗号化に用いる行鍵と列鍵を決定する（ステップＡ１３）。ここで決定された行鍵と列鍵の情報は鍵記憶装置１５に記憶される。

【００３８】

このような設定操作の後、データベースの中の各列項目を順次指定していくと（ステップＡ１４）、その指定された列項目に対する暗号化方式が前記設定情報に基づいて判断される（ステップＡ１５）。この場合、データベースの中の「ｎｕｍｂｅｒ」の列項目については非暗号化項目として設定されているので、何もしない。つまり、「ｎｕｍｂｅｒ」の項目は元データのままである。

【００３９】

また、指定された列項目が検索に利用する列項目として設定されている場合には、鍵記憶装置１５に記憶された当該列項目に共通の列鍵が読み出され（ステップＡ１５→Ａ１６）、当該列項目の各行のデータがその列鍵により暗号化される

（ステップＡ１７）。すなわち、データベースの「ｎａｍｅ」，「ｓｔａｔｅ」，「ａｇｅ」の項目の各行のデータは、図６に示すように、「ａｐｐｌｅ」，「ｏｒａｎｇｅ」，「ｌｅｍｏｎ」といった各列に固有の鍵を用いて暗号化されることになる。

【００４０】

また、指定された列項目が検索に利用する列項目として設定されていない場合、つまり、他の列項目であった場合には、鍵記憶装置１５に記憶された各行に対応した行鍵が読み出され（ステップＡ１５→Ａ１８）、当該列項目の各行のデータがそれぞれに固有の行鍵により暗号化される（ステップＡ１９，Ａ２０）。すなわち、データベースの中の「ｓｔａｔｅ」，「ｗｅｉｇｈｔ」，「ｈｅｉｇｈｔ」の各項目のデータに関しては、図６に示すように、１行目のデータは「ｔｉｇｅｒ」，２行目のデータは「ｄｏｇ」，３行目のデータは「ｃａｔ」，４行目のデータは「ｍｏｕｓｅ」，５行目のデータは「ｅｌｅｐｈａｎｔ」，６行目のデータは「ｃｏｗ」，７行目のデータは「ｐｉｇ」，８行目のデータは「ｒａｂｂｉｔ」，９行目のデータは「ｌｉｏｎ」といったように、各行に対応した行鍵を用いて暗号化される。

【００４１】

このような暗号化処理がデータベースの各列項目について繰り返し行なわれる。すべての列項目の各行のデータの暗号化が終了すると、その暗号化データベースがデータ記憶装置１６のデータベース格納エリア１６ａに上書き保存される（ステップＡ２２）。この状態が図５（ｂ）である。

【００４２】

（ｂ）データベースを検索する場合

図３は本装置にて実行されるデータベース検索処理の動作を示すフローチャートである。今、前記（ａ）で説明した暗号化処理にて、データベースが暗号化されてデータ記憶装置１６に保存されているものとする。

【００４３】

まず、図３（ａ）のフローチャートに示すように、図示せぬデータベース検索用設定画面にて、検索情報の入力を行う（ステップＢ１１）。検索情報の入力と

は、検索対象となる列項目と、検索用の文字列（キーワード）を入力することである。これらの入力情報はデータ記憶装置１６の検索設定情報格納エリア１６ｃに格納される。入力装置１３を通じて検索情報が入力されると、検索前処理が実行される（ステップＢ１２）。

【００４４】

この検索前処理では、図３（ｂ）のフローチャートに示すように、検索対象として入力された列項目が所定の列項目であるか否かが判断され（ステップＣ１１）、所定の列項目であることが判明した場合には（ステップＣ１１のＹｅｓ）、その列項目に共通の列鍵で検索用の文字列が暗号化される（ステップＣ１２）。

【００４５】

所定の列項目とは、前記データベースの暗号化時に設定された検索対象項目（検索に利用される項目）であり、具体的には「ｎａｍｅ」，「ｓｔａｔｅ」，「ａｇｅ」の各項目が該当する。この検索対象項目に関する情報はデータ記憶装置１６の暗号化設定情報格納エリア１６ｂに格納されている。したがって、前記ステップＣ１１では、この暗号化設定情報格納エリア１６ｂを参照して所定の列項目であるか否かの判断を行うことになる。また、列項目に共通の列鍵は鍵記憶装置１５に格納されている。したがって、前記ステップＣ１２では、この鍵記憶装置１５から当該列項目に対応した列鍵を読み出して検索用の文字列を暗号化することになる。例えば、指定項目が「ｓｔａｔｅ」であれば、「ｏｒａｎｇ」といった列鍵を用いて検索用の文字列を暗号化することになる。

【００４６】

また、検索対象として入力された列項目が所定の列項目でなかった場合には（ステップＣ１１のＮｏ）、前記のような検索用文字列の暗号化は行われない。

【００４７】

このような検索前処理の後、データベースの検索処理（図４参照）が行われ（ステップＢ１３）、その検索結果として得られたデータが表示装置１２に表示される（ステップＢ１４）。図４にデータベースの検索処理を示す。

【００４８】

図４は前記ステップＢ１３の検索処理の動作を具体的に示すフローチャートで

ある。

【００４９】

まず、図４（ａ）のフローチャートに示すように、検索用文字列がデータベースとの比較文字列としてデータ記憶装置１６の比較文字列格納エリア１６ｄにセットされる（ステップＤ１１）。この場合、上述したように検索対象として入力された列項目が所定の列項目（「ｎａｍｅ」，「ｓｔａｔｅ」，「ａｇｅ」）であった場合には、前記検索前処理によって、当該検索用文字列がその列項目に対応した列鍵にて暗号化されて比較文字列格納エリア１６ｄにセットされる。その他の列項目の場合には、暗号化されることなく、そのままの状態で比較文字列格納エリア１６ｄにセットされる。

【００５０】

次に、データ記憶装置１６のデータベース格納エリア１６ａに格納された暗号化データベースの列番号による暗号化方式が判断される（ステップＤ１２）。これにより、検索対象が列鍵で暗号化された所定の列項目に該当する場合には、その対象列の各行にあるデータが順次走査され（ステップＤ１２→Ｄ１３）、指定された行に含まれた対象項目のデータの文字列と前記比較文字列格納エリア１６ｄにセットされた検索用文字列（暗号化された文字列）との比較処理が行われる（ステップＤ１４）。

【００５１】

この比較処理では、図４（ｂ）のフローチャートに示すように、データベースから取り出された対象項目のデータの暗号化文字列と検索用の暗号化文字列とを比較して、両者が一致するか否かを判断する（ステップＥ１１）。両者が一致した場合には（ステップＥ１１のＹｅｓ）、その一致した項目を含むレコードデータをデータベース検索結果として抽出する（ステップＥ１２）。

【００５２】

この処理を暗号化データベースの終端まで繰り返して、該当するデータを順次抽出し（ステップＤ１５）、この抽出したデータを検索結果として出力する（ステップＤ２０）。

【００５３】

具体的に説明すると、図５（ｂ）の暗号化データベースの例で、例えば「ｓｔａｔｅ」の項目の中の「Ｆｌｏｒｉｄａ」といったデータを検索することが指定された場合には、まず、検索用データとして入力された「Ｆｌｏｒｉｄａ」を「ｓｔａｔｅ」の列鍵「ａｐｐｌｅ」で暗号化して、「ｈ＊／ｆＤＤ」を得る。この「ｈ＊／ｆＤＤ」といったデータを「ｓｔａｔｅ」の列から検索する。これにより、「ｎｕｍｂｅｒ２」と「ｎｕｍｂｅｒ８」に該当するデータが存在することがわかる。

【００５４】

一方、検索対象が行鍵で暗号化されたその他の列項目に該当する場合には、その対象列の各行にあるデータが順次走査され（ステップＤ１２→Ｄ１６）、指定された行に含まれた対象項目のデータが各行に固有の行鍵によって復号化された後（ステップＤ１７）、前記比較文字列格納エリア１６ｄにセットされた検索用文字列（非暗号化文字列）との比較処理が行われる（ステップＤ１８）。

【００５５】

この比較処理では、図４（ｂ）のフローチャートに示すように、データベースから取り出された対象列のデータの復号化文字列と検索用の非暗号化文字列との比較により、両者が一致するか否かを判断する（ステップＥ１１）。両者が一致した場合には（ステップＥ１１のＹｅｓ）、その一致した項目を含むレコードデータをデータベース検索結果として抽出する（ステップＥ１２）。

【００５６】

この処理を暗号化データベースの終端まで繰り返して、該当するデータを順次抽出し（ステップＤ１９）、この抽出したデータを検索結果として出力する（ステップＤ２０）。

【００５７】

具体的に説明すると、図５（ｂ）の暗号化データベースの例で、例えば「ｗｅｉｇｈｔ」の項目の中の「１６３」といったデータを検索することが指定されたとした場合に、まず、「ｗｅｉｇｈｔ」の１行目のデータを「ｔｉｇｅｒ」といった行鍵で復号化する。同様にして、２行目のデータを「ｄｏｇ」，３行目のデータを「ｃａｔ」，４行目のデータを「ｍｏｕｓｅ」，５行目のデータを「ｅｌ

ｅｐｈａｎｔ」，６行目のデータを「ｃｏｗ」，７行目のデータを「ｐｉｇ」，８行目のデータを「ｒａｂｂｉｔ」，９行目のデータを「ｌｉｏｎ」といったように、各行に対応した行鍵を用いて、それぞれ復号化した後に、検索用データとして入力された「１６３」に基づいて「ｓｔａｔｅ」の列から該当するデータを検索する。これにより、「ｎｕｍｂｅｒ３」と「ｎｕｍｂｅｒ９」に該当するデータが存在することがわかる。

【００５８】

このように、データベースを暗号化する際に、検索に利用される所定の列項目については列共通鍵で暗号化し、検索時には、検索用データをその列共通鍵で暗号化してデータベース上の暗号化データと比較することで高速検索を実現できる。また、所定の列項目以外の列項目については各行毎に異なる鍵を与えて暗号化することでセキュリティを高める。この場合、検索時には各行毎の鍵を用いた復号化を必要とするため、前記所定の列項目に対する検索に比べて時間はかかるものの、検索に頻繁に利用される項目ではないので問題にはならない。

【００５９】

（第２の実施形態）

前記第１の実施形態では、所定の列項目以外の列項目のデータについては、各行毎に固有の行鍵を用いて暗号化するようにしたが、第２の実施形態では、さらにセキュリティを高めるため、各行毎に固有の行鍵と当該列項目に共通の列鍵とを複合的に用いて暗号化することを特徴とする。

【００６０】

図７は第２の実施形態におけるデータベースの構成を示す図であり、図７（ａ）は暗号化前の状態、同図（ｂ）は暗号化後の状態、同図（ｃ）は復号化後の状態を示している。また、図８は第２の実施形態における合成鍵の構成を示す図である。

【００６１】

図７（ａ）に示すように、本装置では、行と列からなるマトリクス形式のデータベースを備えている。ここでは、個人データをデータベース化した場合を示している。このデータベースは、１レコードが「ｎｕｍｂｅｒ」，「ｎａｍｅ」，

「ｓｔａｔｅ」，「ｗｅｉｇｈｔ」，「ｈｅｉｇｈｔ」，「ａｇｅ」，「ｐｈｏｎｅ」の各項目で構成するようにしている。

【００６２】

このようなデータベースに対し、合成鍵を用いて暗号化を行う。すなわち、検索に頻繁に利用される列項目を「ｎａｍｅ」，「ｓｔａｔｅ」，「ａｇｅ」とした場合に、これらの列項目の各行のデータについては、図８に示すように、「ａｐｐｌｅ」，「ｏｒａｎｇｅ」，「ｌｅｍｏｎ」といった各列項目に共通の列鍵を用いて暗号化し、その他の列項目「ｗｅｉｇｈｔ」，「ｈｅｉｇｈｔ」，「ｐｈｏｎｅ」の各行のデータについては、「ｂａｎａｎａ＋行鍵」，「ｌｙｃｈｅｅ＋行鍵」，「ａｐｒｉｃｏｔ＋行鍵」といったように、列鍵と行鍵とを組み合わせて暗号化する。

【００６３】

なお、「ｎｕｍｂｅｒ」の列は暗号化を行わないものとする。行鍵としては、「ｔｉｇｅｒ」，「ｄｏｇ」，「ｃａｔ」，「ｍｏｕｓｅ」，「ｅｌｅｐｈａｎｔ」，「ｃｏｗ」，「ｐｉｇ」，「ｒａｂｂｉｔ」，「ｌｉｏｎ」といった鍵が用いられる。これらの鍵は所定の関数を用いて各列または各行毎に数学的に発生させることができる。

【００６４】

図７（ａ）のデータベースを合成鍵を用いて暗号化した結果を図７（ｂ）に示す。データ記憶装置１６のデータベース格納エリア１６ａには、図７（ｂ）に示すような状態でデータベースが保存されることになる。

【００６５】

データベースを検索する場合に、前記第１の実施形態と同様に、検索に利用される列項目に共通の列鍵を用いて検索用データを暗号化してから検索処理を行えば良い。例えば、「ｓｔａｔｅ」の中の「Ｆｌｏｒｉｄａ」といったデータを検索する場合には、まず、検索用データとして入力された「Ｆｌｏｒｉｄａ」を「ｓｔａｔｅ」の列鍵「ａｐｐｌｅ」で暗号化して、「ｈ＊／ｆＤＤ」を得る。この「ｈ＊／ｆＤＤ」といったデータを「ｓｔａｔｅ」の列の各行から検索する。これにより、「ｎｕｍｂｅｒ２」と「ｎｕｍｂｅｒ８」に該当するデータが存在

することがわかる。

【００６６】

また、暗号化されたデータベースを元に戻す場合には、暗号化の時と同じ合成鍵を用いる。図７（ｂ）のデータベースを暗号化の時と同じ合成鍵を用いて復号化すると、図７（ｃ）に示すように元のデータを得ることができる。

【００６７】

なお、データベースを暗号化する場合の処理や、暗号化されたデータベースを検索する場合の処理については、所定の列項目以外の列項目の各行のデータに対して列鍵と行鍵を組み合わせて暗号化する点を除いては前記第１の実施形態の処理（図２～図４）と同様であるため、ここでは、その説明を省略するものとする。

【００６８】

このように、検索に頻繁に利用される列項目については、その列項目に共通の列鍵を用いて暗号化することで、前記第１の実施形態と同様に高速検索を実現できるものであり、また、他の列項目については、列鍵と行鍵とを複合的に用いて暗号化することで、さらにセキュリティを強化することができる。

【００６９】

（第３の実施形態）

前記第１または第２の実施形態では、本発明を装置単体で構成したが、データベースの保管場所を離間させておき、別の情報端末からネットワークを介して検索を依頼するようなデータベースシステムを構築することも可能である。

【００７０】

以下に、このようなデータベースシステムについて説明する。

【００７１】

図９は本発明の第３の実施形態に係るデータベースシステムの構成を示すブロック図である。本システムは、第１の端末装置２０と第２の端末装置３０とを有する。第１の端末装置２０と第２の端末装置３０とはネットワーク４０を介して接続されている。

【００７２】

第１の端末装置２０は、データベースサービスを行うサーバコンピュータとして用いられるものであって、データベースの検索処理を行う検索装置２１と、データベースを保存しておくためのデータ記憶装置２２とで構成される。第２の端末装置３０は、データベースの検索を第１の端末装置２０に依頼し、その結果を第１の端末装置２０から受け取るクライアントコンピュータとして用いられるものであって、検索依頼装置３１と復号化装置３２とで構成される。

【００７３】

このようなデータベースシステムにあっては、第１の端末装置２０において、図２で説明したように、データベースの所定の列項目の各行のデータを当該列項目に共通の列鍵を用いて暗号化し、その他の列項目の各行のデータについては各行毎に固有の行鍵を用いて暗号化してデータ記憶装置２２に保存しておくものとする。

【００７４】

ここで、第２の端末装置３０から第１の端末装置２０に対してデータベースの検索を依頼する際に、図３の検索前処理までを第２の端末装置３０側で実行する。すなわち、第２の端末装置３０の検索依頼装置３１によって、検索対象として入力された列項目が所定の列項目であるか否かを判断し、所定の列項目である場合に、検索用の文字列（キーワード）を当該列項目に共通の列鍵を用いて暗号化する。その他の列項目が検索対象である場合には、このような暗号化は不要である。

【００７５】

検索前処理の後、第２の端末装置３０から検索用文字列をネットワーク４０を介して第１の端末装置２０に送る。第１の端末装置２０側では、この検索用文字列を受けることにより、図４で説明したような検索処理を実行する。

【００７６】

すなわち、第１の端末装置２０の検索装置２１により、検索対象が所定の列項目か否かを判断し、所定の列項目であれば、第２の端末装置３０から取得した検索用文字列（暗号化文字列）とデータ記憶装置２２内の暗号化データベースの当該列項目の各行のデータとを比較して、該当するデータを抽出するといった処理

を行う。また、検索対象が所定の列項目以外の列項目であれば、データ記憶装置２２内の暗号化データベースの当該列項目のデータを各行毎の鍵で復号化してから、第２の端末装置３０から取得した検索用文字列（非暗号化文字列）とその復号化された各行のデータとを比較して、該当するデータを抽出するといった処理を行う。

【００７７】

このようにして検索結果が得られると、第１の端末装置２０はその検索結果として得られたデータを暗号化の状態のままでネットワーク４０を介して第２の端末装置３０に返信する。第２の端末装置３０は、第１の端末装置２０と共通の暗号鍵を持っている。したがって、第１の端末装置２０から検索結果を受信すると、内部の復号化装置３２にて、そのデータを暗号鍵を用いて復号化することができる。この場合、第１の端末装置２０と第２の端末装置３０との間で、常に暗号化された状態でデータがやり取りされるので、データベースのセキュリティを確保することができる。

【００７８】

このように、第１の端末装置２０側にデータベースを持たせて、第２の端末装置３０からのアクセスによってデータベース検索を行うようにしたデータベースシステムであっても、検索に頻繁に使う列項目のデータを当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては各行毎に固有の行鍵を用いて暗号化しておくことで、セキュリティを高めると共に高速検索を実現することができる。

【００７９】

なお、前記所定の列項目以外の列項目のデータについては、前記第２の実施形態で説明したように、各行毎に固有の行鍵と当該列項目に共通の列鍵とを複合的に用いて暗号化することでも良く、このようにすれば、さらにセキュリティを強化することができる。

【００８０】

（暗号化方式）

次に、本発明に適用されるデータベースの暗号化方式について説明する。

【００８１】

本発明では、データベースの暗号化アルゴリズムとして多次元空間回転方式（多次元空間ベクトル方式）を用いる。この多次元空間回転方式は、所定の関数に基づいて多次元空間に逐次ベクトルを発生させ、これらのベクトル成分を暗号のキーストリームとするものである。この多次元空間回転方式では、演算能力の低い情報処理装置でも計算できるので、携帯型の端末に適用するのに相応しい。つまり、本発明のデータベースを外部からアクセスするような環境において、データの機密性を保持して処理するためには、このような暗号化方式を採用することが望ましい。

【００８２】

以下に、図１０および図１１を参照して、多次元空間回転方式を用いた暗号化と復号化の処理を示す。

【００８３】

図１０（ａ）は多次元空間回転方式を用いた場合での暗号化の処理動作を示すフローチャートである。

【００８４】

暗号化の対象となる原データをＭとする（ステップＦ１１）。このデータＭは、バイナリデータである。まず、このデータＭに対してｂｉｔ単位でスクランブル１をかける（ステップＦ１２）。こうして得られたデータをＭ´とする（ステップＦ１３）。

【００８５】

ここで、このデータＭ´に数学的に順次生成される乱数をＸＯＲ（排他的論理和）して暗号化を行う（ステップＦ１４）。このとき、乱数の発生の関数に多次元ベクトルｒを用いることが多次元空間回転方式の特徴である。この場合、多次元ベクトルｒを発生させる関数、あるいは、その関数に用いられる定数は暗号化キー（秘密・公開キー）で決定される。

【００８６】

すなわち、暗号化に際し、秘密キー（P_1，P_2）および公開キー（P_3，P_4）を定数として用いた関数に従って多次元ベクトルｒを生成し、Ｍ´XOR ｒと

いった論理演算を行ってデータＭ′を暗号化する。こうして得られた暗号データをＣとする（ステップＦ１５）。

【００８７】

具体的に説明すると、例えば図１１に示すように、ｒが３次元ベクトル（ｘ，ｙ，ｚ）であり、そのベクトル成分ｘ，ｙ，ｚの計算精度が１６ｂｉｔであるとする。この３次元ベクトルｒ（ｘ，ｙ，ｚ）を後述する式（１）に従ってr_0，r_1，r_2，r_3…といったように順次発生する。

【００８８】

今、データＭが８ｂｉｔデータの並びとして、m_0 m_1 m_2 m_3 m_4 m_5 m_6…と与えられたとき、Ｍは前記計算精度（１６ｂｉｔ）に基づいて、その要素の２つ（８ｂｉｔ）ずつに分解される。そして、３次元ベクトルがr_0である場合には、データＭとr_0（x_0，y_0，z_0）とのＸＯＲ（排他的論理和）により、（ｘ XOR m_0 m_1）（ｙ XOR m_2 m_3）（ｚ XOR m_4 m_5）…といった計算が行われ、この計算結果として、C_0 C_1 C_2 C_3 C_4 C_5…といった暗号データＣが得られる。

【００８９】

このようにして得られたデータＣに対して、さらにｂｉｔ単位でスクランブル２をかける（ステップＦ１６）。こうして得られたデータをＣ′とし、最終的な暗号データとして出力する（ステップＦ１７）。

【００９０】

なお、前記の処理で、Ｃ′を改めてＭ′と見なして前記同様の暗号化を繰り返すことによって、復号化の困難さのレベルを上げることができる。また、このときの多次元ベクトルｒを生成する関数の形を変えれば、復号化の困難さはさらに増すことになる。

【００９１】

次に、復号化の処理動作について説明する。

【００９２】

図６（ｂ）は復号化の処理動作を示すフローチャートである。

【００９３】

復号化は、基本的には前記暗号化処理の逆の処理を行えば良い。

【００９４】

すなわち、暗号データをＣ′とすると（ステップＧ１１）、まず、このデータＣに対してｂｉｔ単位で前記暗号化時に行ったスクランブル２とは逆のスクランブル２をかける（ステップＧ１２）。これにより、スクランブル２をかける前のデータＣが得られる（ステップＧ１３）。

【００９５】

次に、Ｃ XOR ｒといった計算を行ってデータＣを復号化する（ステップＧ１４）。これにより、暗号化を行う前のデータＭ′が得られる（ステップＧ１５）。

【００９６】

そして、このデータＭ′に対してもｂｉｔ単位で前記暗号化時に行ったスクランブル１とは逆のスクランブル１をかける（ステップＧ１２）。これにより、スクランブル１をかける前のデータつまり原データＭが得られる（ステップＧ１７）。

【００９７】

なお、暗号化の処理で、Ｃ′を改めてＭ′と見なして暗号化を繰り返したり、多次元ベクトルｒを生成する関数の形を変えたりする処理が加わっていれば、その処理に対応させて復号化の処理を行うものとする。

【００９８】

ところで、多次元ベクトルｒを用いた暗号化において、多次元ベクトルｒを扱う関数を決定するパラメータ（定数）の集合Ｐを２つの部分に分けて、

$$P = \{P_s, P_p\}$$

と表わす。ここで、P_s は秘密パラメータであり、暗号化キー（秘密キーP_1，P_2）に相当する。P_p は公開パラメータであり、暗号化キー（公開キーP_3，P_4）に相当する。

【００９９】

次に、多次元空間回転方式について、さらに詳しく説明する。

【０１００】

ｎ（ｎ≧１）次元の空間を張るベクトルをｒとし、その初期値r_0から逐次的に新しいベクトルr_i（ｉ＝０，１，２，３…）を発生する関数をＲとする。このとき、r_iは以下のような式（１）で表わされる。

【０１０１】

$$r_i = A \cdot R(P, r_{i-1})\, r_{i-1} + c \qquad \cdots (1)$$

ここで、Ａは適当な定数係数である。Ｐは関数に使用される定数の集合であり、秘密キーP_1，P_2と、公開キーP_3，P_4が使用される。ｃはベクトルを並進移動する定数ベクトルである。

【０１０２】

前記式（１）において、係数Ａは関数Ｒに適当な制約（例えば｜Ｒ｜≦１）を設けたときに、各ベクトルが多次元空間の閉空間領域内に存在するための条件を与える。定数ベクトルｃはベクトルr_iが収斂するとき、「トリビアル」な点（例えばｒ＝０のような無意味な点）にならないことを保証する（勿論ｃ＝０も許される）。

【０１０３】

ｎ次元空間では、ベクトルｒはｎ個の要素を持つ。ｒ＝（x_1，x_2 …x_n）。計算機上では、数値データ一般に、コンパイラが定義するビット長（ｍ）の精度（例えば８バイト又は６４ビット）で表現されている。したがって、ベクトルの逐次生成法のある瞬間にｎ×ｍのデータの精度でベクトルｒを再現できないと、それに続くベクトルｒは正確には再現できない（あるいはそうなるような関数Ｒを定義する）。これは、ベクトルｒの初期値r_0についても同様で、初期値r_0をｎ×ｍのデータの精度で再現したときにのみ、それに続くベクトルr_1，r_2，r_3…の再現性が保証される。

【０１０４】

前記式（１）を用いて得られたベクトルｒの成分を、そのデータ定義長に応じて１個乃至複数個並べて、その総ビット長に対応した文字列（１文字あたり８ビットが普通）とビット毎の排他論理演算（ＸＯＲ）を行う。これを第１暗号化とする。これについては、前記図１１で述べた通りある。

【０１０５】

この手続きは暗号解読への対策となる。この場合、再度式（１）、関数Ｒを変えてて新しいベクトルを発生させ、第１暗号化と同じ手法により暗号化を行っても良い。これを第２暗号化とする。

【０１０６】

具体的な例として、ｎ＝２の場合を考える。

【０１０７】

まず、r_{i-1} をこの平面に立てた法線の周りにθだけが回転する演算として、Ｒを定義すれば、Ｒは２×２のマトリックスとなり、次のように表わされる。

【０１０８】

【数１】

$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \qquad \cdots (2)$$

【０１０９】

この場合、θは一種の媒介変数となっている。すなわち、媒介変数が r_{i-1} の関数として与えられ、

$$\theta (r) = f (P, r) \qquad \cdots (3)$$

で表わされるとき、前記式（２）で表わされる変換は、形式的に前記式（１）で表わされる。

【０１１０】

なお、前記式（３）において、Ｐは関数ｆに使われる定数の集合として定義されるものであり、秘密キーP_1，P_2と、公開キーP_3，P_4が使用される。

【０１１１】

このように、逐次的に発生させた多次元空間を張るベクトルｒを暗号化に使うことで、ＲＳＡのような暗号化に比べ、計算機の処理精度や処理能力に依存しない暗号化を実現できる。

【０１１２】

さて、このような多次元空間回転方式を本発明のデータベースの暗号化に適用する場合には、行鍵と列鍵を

$$r' = A \cdot R (P, r)\ r + c \qquad \cdots (4)$$

の関数ＲのＰとして利用する。前記式（４）のＰは関数Ｒに使用される定数の集合である。本発明の場合には、「ａｐｐｌｅ」，「ｏｒａｎｇｅ」，「ｌｅｍｏｎ」等を秘密キーとして用いる。すなわち、前記の説明では、秘密キーとしてP_1，P_2と定めているので、例えば「ａｐｐｌｅ」を秘密キーとする場合には、P_1を「ａｐ」、P_2を「ｐｌｅ」とする。なお、これは一例であり、鍵の設定の仕方はこれに限らず、いろいろな組み合わせが考えられる。

【０１１３】

このように、多次元空間回転方式を本発明のデータベースの暗号化に適用する場合には、行鍵と列鍵を多次元空間回転方式の所定の関数の定数成分として用いる。これにより、高精度の演算能力を必要とせずに、解読困難な暗号結果を得ることができる。

【０１１４】

なお、多次元空間回転方式では、次元数が大きくなると、回転行列Ｒ（関数）の要素も多くなり、暗号化／復号化時の演算負荷が大きくなる問題がある。このような問題を解消する方法として、多次元空間回転方式を用いた暗号化方式における多次元空間回転行列を次数の少ない疑似空間回転行列に置き換えて計算する方法がある。

【０１１５】

具体的に説明すると、例えば、６次元空間における回転行列Ｒの要素は、式（５）のようになり、計算量が膨大に増える。

【０１１６】

【数２】

$$R = \begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} & R_{1,4} & R_{1,5} & R_{1,6} \\ R_{2,1} & R_{2,2} & R_{2,3} & R_{2,4} & R_{2,5} & R_{2,6} \\ R_{3,1} & R_{3,2} & R_{3,3} & R_{3,4} & R_{3,5} & R_{3,6} \\ R_{4,1} & R_{4,2} & R_{4,3} & R_{4,4} & R_{4,5} & R_{4,6} \\ R_{5,1} & R_{5,2} & R_{5,3} & R_{5,4} & R_{5,5} & R_{5,6} \\ R_{6,1} & R_{6,2} & R_{6,3} & R_{6,4} & R_{6,5} & R_{6,6} \end{bmatrix} \quad \cdots (5)$$

【０１１７】

そこで、このような回転行列Ｒを疑似回転行列Ｑに置き換えて計算する。疑似

回転行列Ｑは、次数の少ない空間回転行列の要素を対角に配置し、残りの要素を０とした擬似的な行列である。例えば、６次元空間の場合には、式（６）のような疑似回転行列Ｑを用いる。

【０１１８】

【数３】

$$Q = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} = \begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} & 0 & 0 & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} & 0 & 0 & 0 \\ A_{3,1} & A_{3,2} & A_{3,3} & 0 & 0 & 0 \\ 0 & 0 & 0 & B_{1,1} & B_{1,2} & B_{1,3} \\ 0 & 0 & 0 & B_{2,1} & B_{2,2} & B_{2,3} \\ 0 & 0 & 0 & B_{3,1} & B_{3,2} & B_{3,3} \end{bmatrix} \quad \cdots (6)$$

【０１１９】

なお、式（６）において、Ａ，Ｂは３次元回転行列である。

【０１２０】

この疑似回転行列Ｑと前記回転行列Ｒの要素を比較して見ると、Ｑの要素には０が多く、計算量が少なくて済む。しかも、暗号化としての機能も十分に果たす。一般的には、式（７）のように、多次元空間回転行列Ｑを設定すれば良い。

【０１２１】

【数４】

$$Q = \begin{bmatrix} A_1 & 0 & \cdot & \cdot & 0 \\ 0 & A_2 & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & A_i \end{bmatrix} \quad \cdots (7)$$

（$A_1, A_2 \cdots, A_i$は多次元空間回転行列）

【０１２２】

このように、多次元空間回転方式における回転行列を次元数の少ない複数の回転行列の要素を対角に配置し、残りを０要素とした疑似回転行列に置き換えて計算することで、計算量を大幅に減らして速やかに暗号化または復号化の処理を実行することができる。

【０１２３】

また、別の方法として、

$$P = S \cdot Q \cdot S^{T} \qquad \cdots (8)$$

として得られるＰを新たな疑似空間回転行列として用いても良い。

【０１２４】

式（８）において、Ｑは前記疑似回転行列である。Ｓは置換行列であり、式（９）のように、各行各列に１の要素が１つ含まれた正方行列である。

【０１２５】

【数５】

$$S = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad \cdots (9)$$

【０１２６】

例えば、疑似回転行列Ｑが前記式（６）で表現される場合（６次元の場合）には、疑似空間回転行列Ｐは、式（１０）のように表せる。

【０１２７】

【数６】

$$P = \begin{bmatrix} A_{1,1} & 0 & A_{1,2} & 0 & A_{1,3} & 0 \\ 0 & B_{1,1} & 0 & B_{1,2} & 0 & B_{1,3} \\ A_{2,1} & 0 & A_{2,2} & 0 & A_{2,3} & 0 \\ 0 & B_{2,1} & 0 & B_{2,2} & 0 & B_{2,3} \\ A_{3,1} & 0 & A_{3,2} & 0 & A_{3,3} & 0 \\ 0 & B_{3,1} & 0 & B_{3,2} & 0 & B_{3,3} \end{bmatrix} \qquad \cdots (10)$$

【０１２８】

これは、置換行列Ｓが式（１１）のような場合である。

【０１２９】

【数７】

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \cdots (11)$$

【０１３０】

このように、多次元空間回転方式における回転行列を次元数の少ない複数の回転行列の要素を対角に配置し、残りを０要素とした疑似回転行列に置換行列を組み合わせてできる新たな疑似回転行列に置き換えて計算すれば、計算の行程が複雑化するため、暗号の解読をさらに困難なものにできる。

【０１３１】

【発明の効果】

以上詳記したように本発明によれば、データベースを暗号化する際に、検索に利用される列項目のデータについては当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては各行毎に固有の行鍵を用いて暗号化するようにしたため、各行毎に鍵を異ならせることでセキュリティを高めることができ、検索時には、検索用として入力されたデータを前記所定の列項目に共通の列鍵を用いて暗号化し、その暗号化された検索用データと暗号化データベースとを比較することで高速検索を実現することができる。

【図面の簡単な説明】

【図１】

本発明の第１の実施形態に係るデータベース管理装置の構成を示す図。

【図２】

前記データベース管理装置にて実行されるデータベース暗号化処理の動作を示すフローチャート。

【図３】

前記データベース管理装置にて実行されるデータベース検索処理の動作を示すフローチャート。

【図４】

前記図３のステップＢ１３の検索処理の動作を具体的に示すフローチャート。

【図５】

第１の実施形態におけるデータベースの構成を説明するための図であり、図５（ａ）は暗号化前の状態、同図（ｂ）は暗号化後の状態、同図（ｃ）は復号化後の状態を示す図。

【図６】

第１の実施形態における列鍵と行鍵の構成を示す図。

【図７】

本発明の第２の実施形態におけるデータベースの構成を示す図であり、図７（ａ）は暗号化前の状態、同図（ｂ）は暗号化後の状態、同図（ｃ）は復号化後の状態を示す図。

【図８】

第２の実施形態における合成鍵の構成を示す図。

【図９】

本発明の第３の実施形態に係るデータベースシステムの構成を示すブロック図。

【図１０】

多次元空間回転方式を用いた場合での暗号化処理と復号化処理の動作を示すフローチャート。

【図１１】

多次元空間回転方式を用いた場合での暗号化の演算方法を説明するための図。

【符号の説明】

１１…ＣＰＵ

１２…表示装置

１３…入力装置

１４…プログラム記憶装置

１５…鍵記憶装置

１６…データ記憶装置

１６ａ…データベース格納エリア

１６ｂ…暗号化設定情報格納エリア

１６ｃ…検索設定情報格納エリア

１６ｄ…比較文字列格納エリア

２０…第１の端末装置

２１…検索装置

２２…データ記憶装置

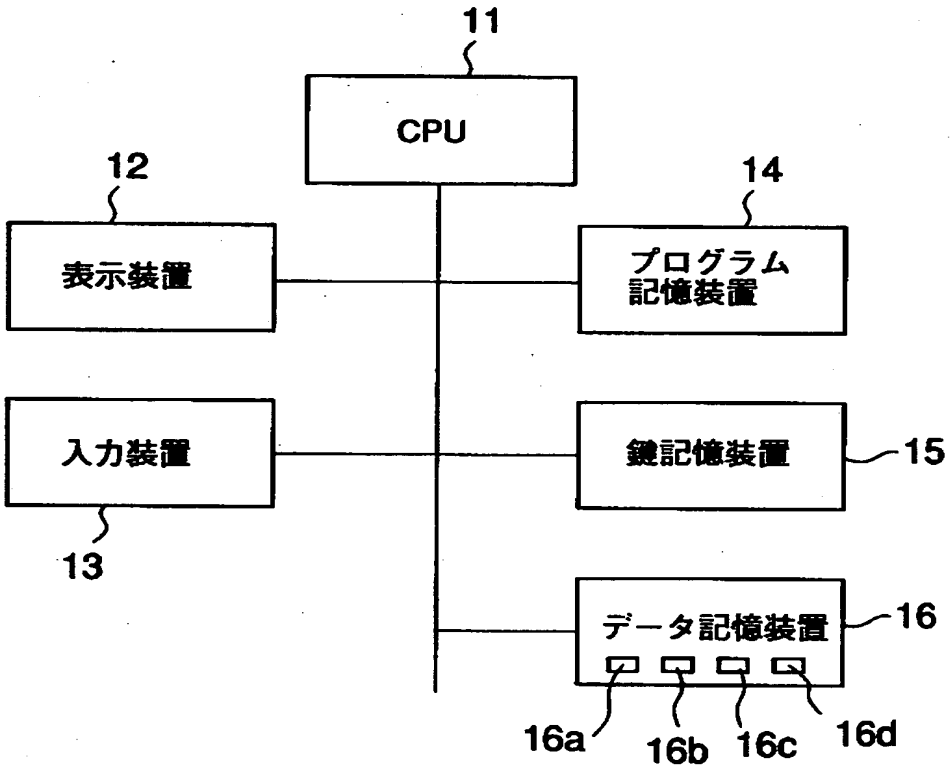
３０…第２の端末装置

３１…検索依頼装置

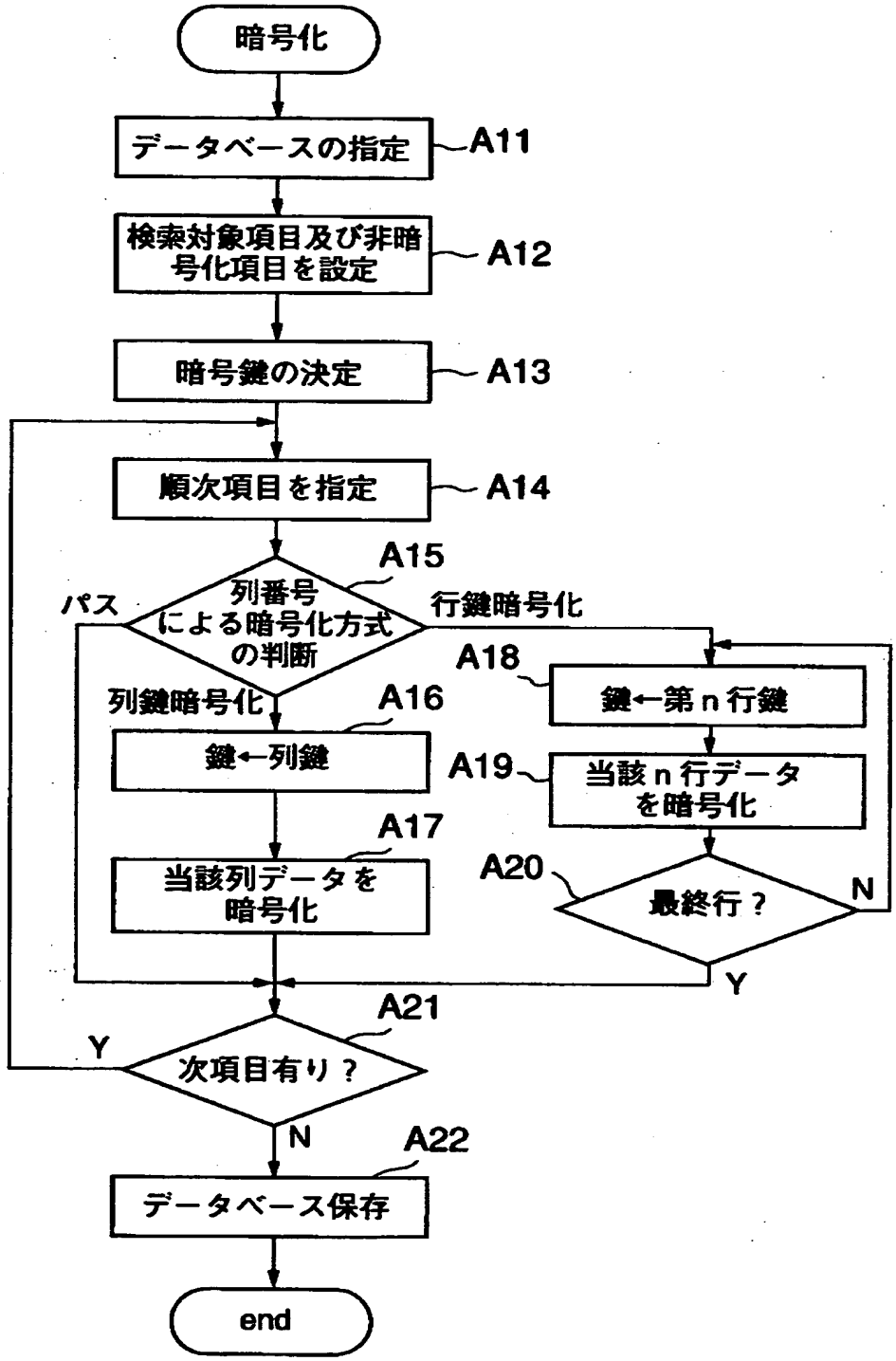
３２…復号化装置

【書類名】　　図面

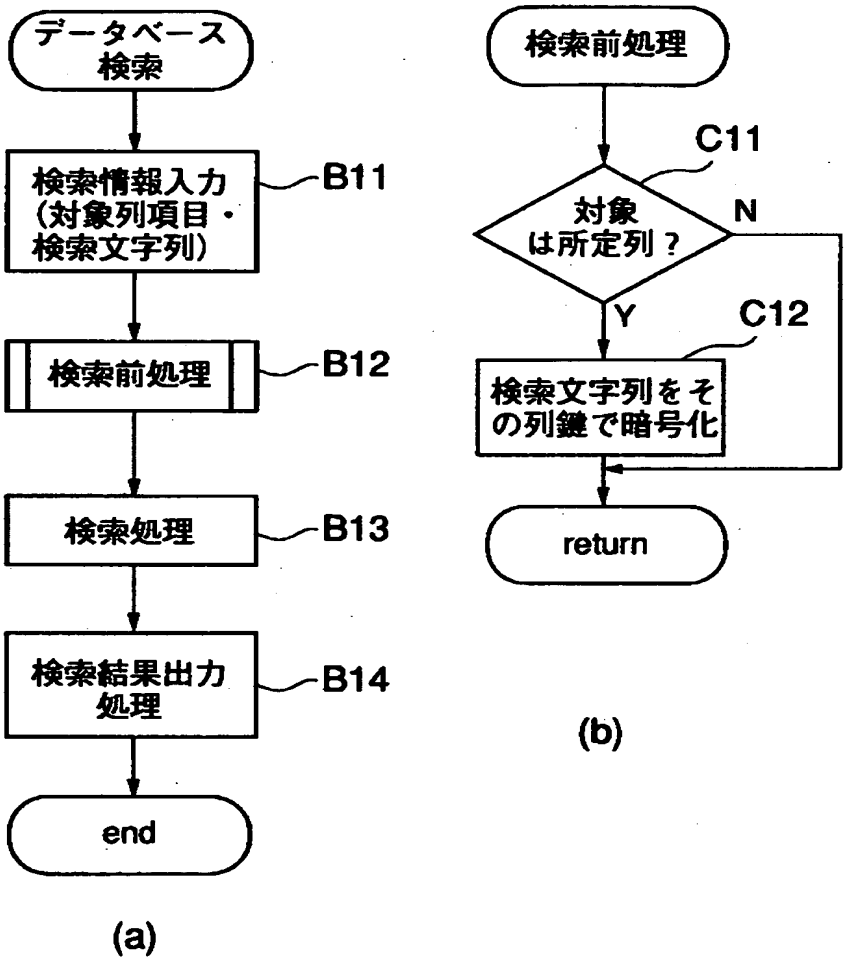
【図１】


11
CPU
12
表示装置
14
プログラム
記憶装置
入力装置
13
鍵記憶装置
15
データ記憶装置
16
16a
16b
16c
16d

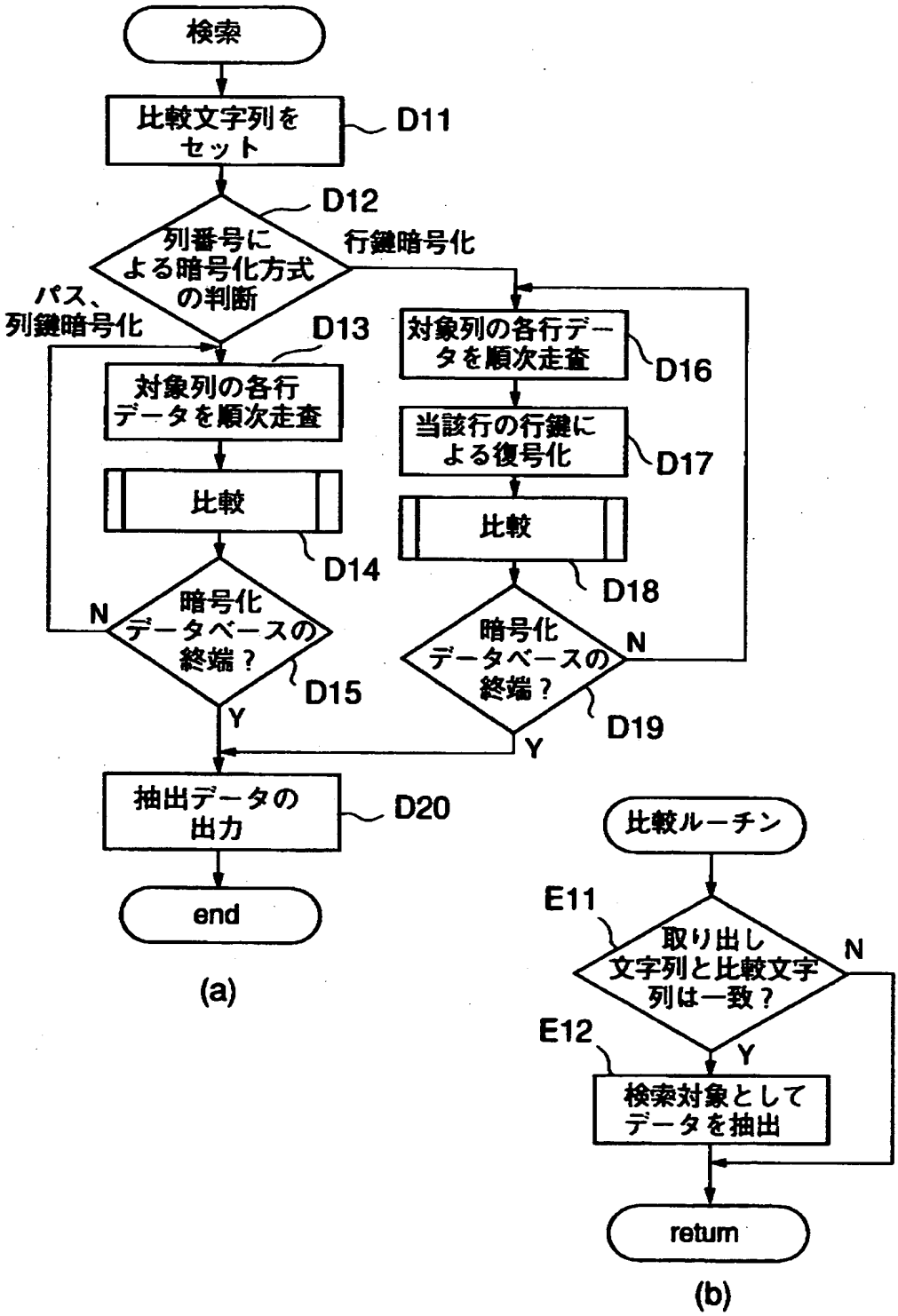
【図２】


暗号化
データベースの指定
A11
検索対象項目及び非暗号化項目を設定
A12
暗号鍵の決定
A13
順次項目を指定
A14
A15
列番号による暗号化方式の判断
パス
行鍵暗号化
列鍵暗号化
A16
鍵←列鍵
A17
当該列データを暗号化
A18
鍵←第ｎ行鍵
A19
当該ｎ行データを暗号化
A20
最終行？
N
Y
A21
次項目有り？
Y
N
A22
データベース保存
end

【図３】


データベース検索
検索情報入力（対象列項目・検索文字列）
B11
検索前処理
B12
検索処理
B13
検索結果出力処理
B14
end
(a)
検索前処理
C11
対象は所定列？
N
Y
C12
検索文字列をその列鍵で暗号化
return
(b)

【図４】


検索
比較文字列をセット
D11
D12
列番号による暗号化方式の判断
行鍵暗号化
パス、列鍵暗号化
D13
対象列の各行データを順次走査
比較
D14
暗号化データベースの終端？
N
Y
D15
対象列の各行データを順次走査
D16
当該行の行鍵による復号化
D17
比較
D18
暗号化データベースの終端？
N
Y
D19
抽出データの出力
D20
end
(a)
比較ルーチン
E11
取り出し文字列と比較文字列は一致？
N
Y
E12
検索対象としてデータを抽出
return
(b)

【図５】

(a)

number	name	state	weight	height	age	phone
1	Jhon	New York	63	130	22	407-228-6611
2	Chris	Florida	72	190	21	123-456-7890
3	Michael	Minnesota	65	163	27	101-202-3030
4	David	Iowa	63	145	34	523-761-0045
5	Mark	New York	65	152	30	832-962-9001
6	Daniel	Iowa	68	170	25	231-981-9454
7	George	Idaho	69	180	31	561-545-4389
8	Henry	Florida	71	165	22	239-203-9800
9	Joe	New Jersey	66	163	27	239-129-9898

暗号化 列鍵 行鍵

(b)

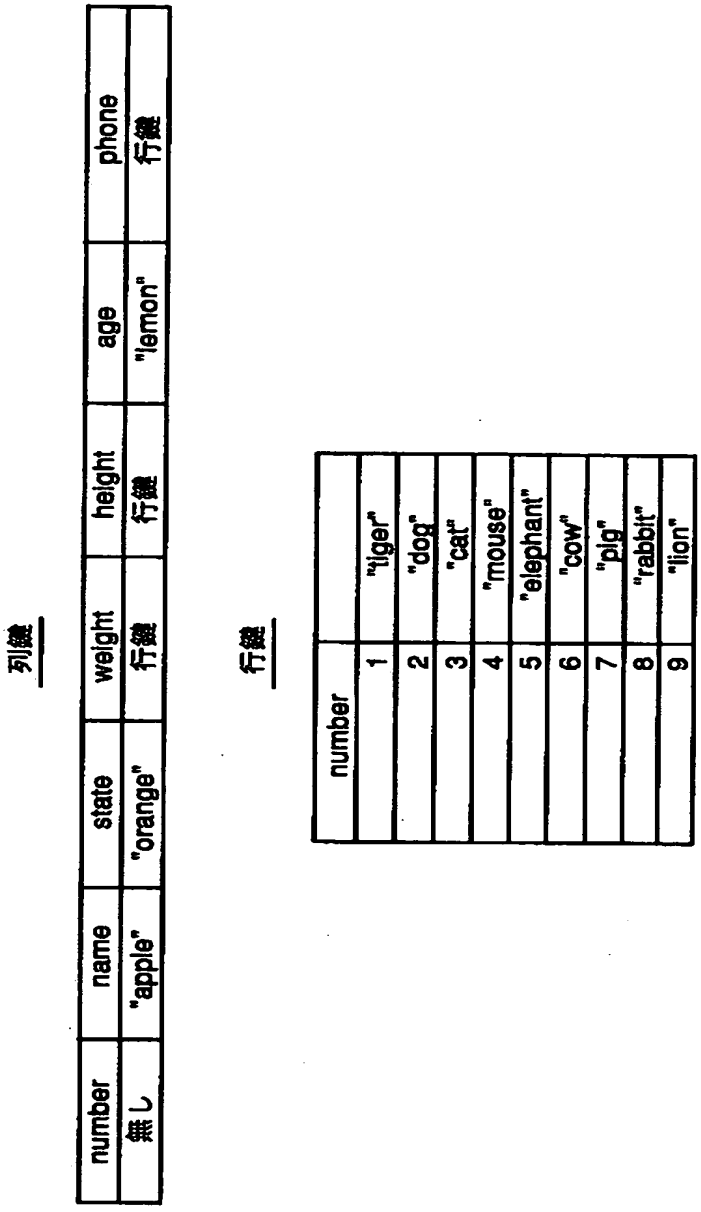
number	name	state	weight	height	age	phone
1	wJls	noevjolc	qw	ywe	jh	igdltytfhDSk
2	ddGGa	h*/fDD	lr	Erw	hg	LKtYfDSkoKow
3	1jkjl+P	gah{6xpVd	RK	Tyj	tY	hkliiydageQk
4	3eK@s	kHHS	kd	DHH	Kl	d+fDIKnBerJf
5	erlN	noevjolc	jD	iOO	Gv	wsdERfvW2Sdf
6	F>sSlu	kHHS	8u	lkj	ij	1xcVlmFmkjpo
7	{:ld?k	IJHFD	HH	lpa	LK	kjwDkJGvfDoa
8	rhJKd	h*/fDD	ew	Aij	jh	e419h-ka+qwH
9	ifd	ASoChijlO-	Df	lky	tY	qLFUiCVkj@kl

復号化 列鍵 行鍵

(c)

number	name	state	weight	height	age	phone
1	Jhon	New York	63	130	22	407-228-6611
2	Chris	Florida	72	190	21	123-456-7890
3	Michael	Minnesota	65	163	27	101-202-3030
4	David	Iowa	63	145	34	523-761-0045
5	Mark	New York	65	152	30	832-962-9001
6	Daniel	Iowa	68	170	25	231-981-9454
7	George	Idaho	69	180	31	561-545-4389
8	Henry	Florida	71	165	22	239-203-9800
9	Joe	New Jersey	66	163	27	239-129-9898

【図６】


列鍵
number
name
state
weight
height
age
phone
無し
"apple"
"orange"
行鍵
行鍵
"lemon"
行鍵
行鍵
number
1
"tiger"
2
"dog"
3
"cat"
4
"mouse"
5
"elephant"
6
"cow"
7
"pig"
8
"rabbit"
9
"lion"

【図７】

(a)

number	name	state	weight	height	age	phone
1	Jhon	New York	63	130	22	407-228-6611
2	Chris	Florida	72	190	21	123-456-7890
3	Michael	Minnesota	65	163	27	101-202-3030
4	David	Iowa	63	145	34	523-761-0045
5	Mark	New York	65	152	30	832-962-9001
6	Daniel	Iowa	68	170	25	231-981-9454
7	George	Idaho	69	180	31	561-545-4389
8	Henry	Florida	71	165	22	239-203-9800
9	Joe	New Jersey	66	163	27	239-129-9898

暗号化　合成鍵

(b)

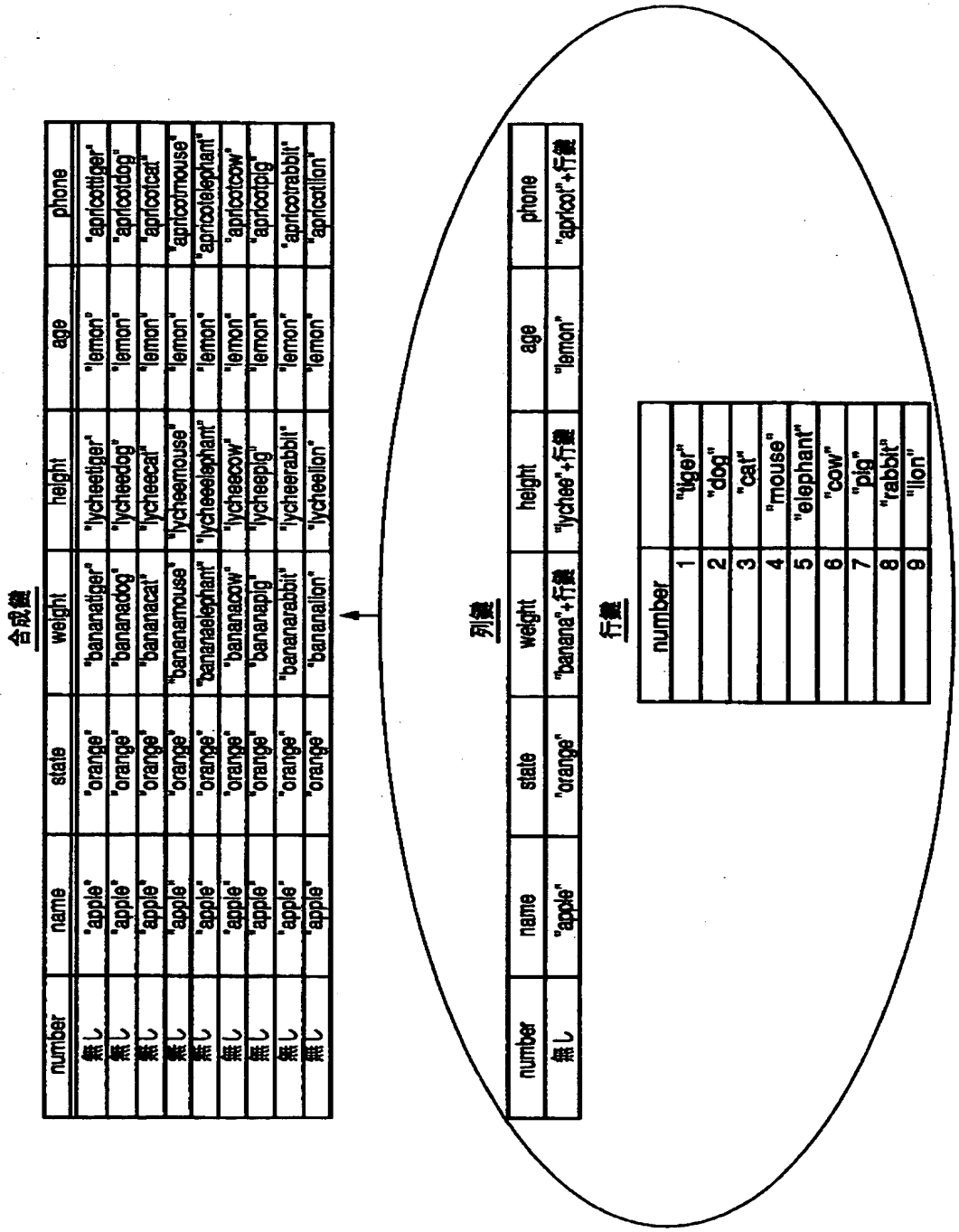
number	name	state	weight	height	age	phone
1	wJls	noevjolc	xo	qwe	jh	dfghaJ;lkqlu
2	ddGGa	h*/fDD	wi	klA	hg	qwTylBnDFiKj
3	1jkjl+P	gah{6xpVd	hj	lKJ	tY	DafgiqlkimD—
4	3eK@s	kHHS	s?	SGA	Kl	hi"khaTygfXd
5	erlN	noevjolc	d-	ASD	Gv	8uyDBmAkolka
6	f>sSlu	kHHS	l*	qoK	ij	jhtvbnMKJASW
7	):ld?k	lJHFD	df	sLL	LK	lQwSRyuiokjq
8	rhJKd	h*/fDD	Ws	tyH	of	Dfha*kaqil
9	ifd	ASoChijlO-	qo	H2a	Ga	lkjHYAGoiuq

復号化　合成鍵

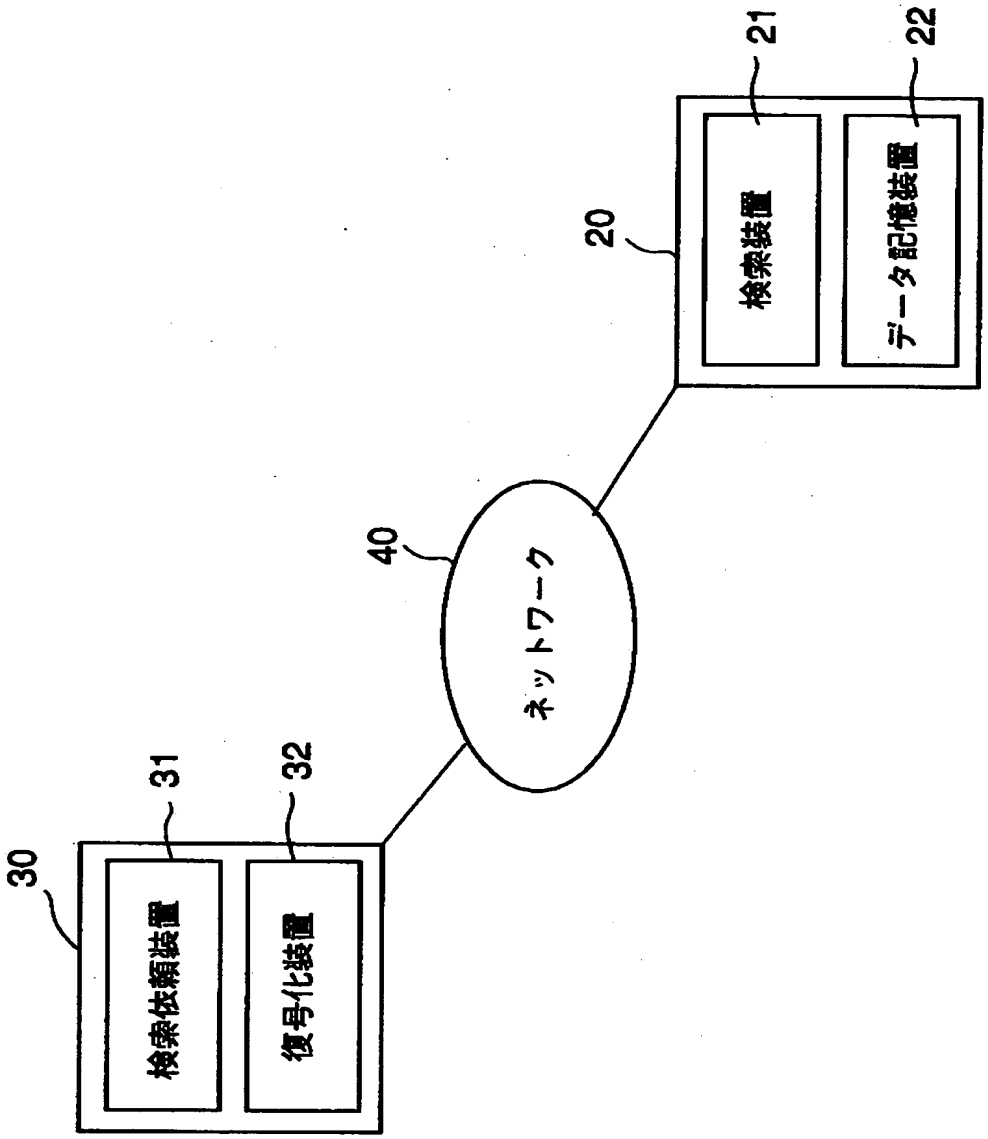
(c)

number	name	state	weight	height	age	phone
1	Jhon	New York	63	130	22	407-228-6611
2	Chris	Florida	72	190	21	123-456-7890
3	Michael	Minnesota	65	163	27	101-202-3030
4	David	Iowa	63	145	34	523-761-0045
5	Mark	New York	65	152	30	832-962-9001
6	Daniel	Iowa	68	170	25	231-981-9454
7	George	Idaho	69	180	31	561-545-4389
8	Henry	Florida	71	165	22	239-203-9800
9	Joe	New Jersey	66	163	27	239-129-9898

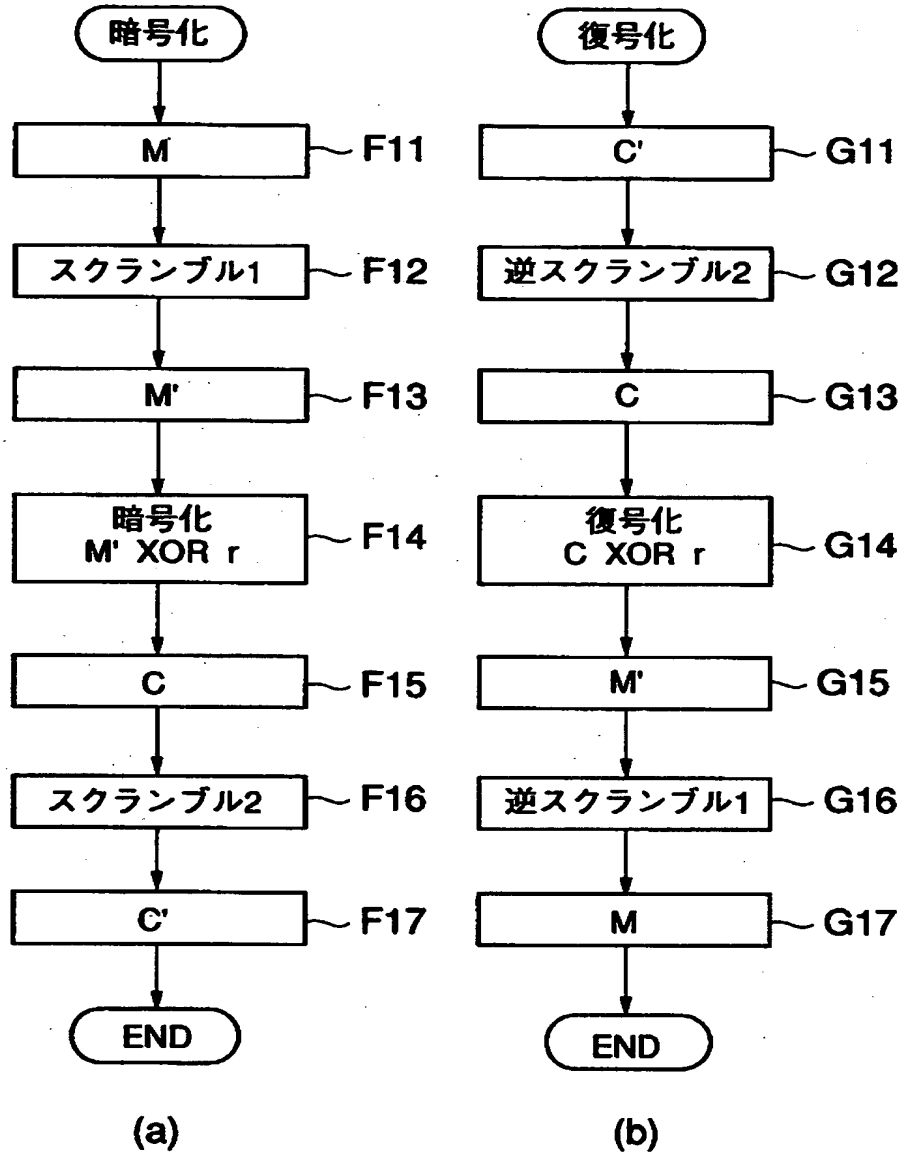
【図８】


合成量
number
name
state
weight
height
age
phone
無し
"apple"
"orange"
"bananatiger"
"lycheetiger"
"lemon"
"apricottiger"
"bananadog"
"lycheedog"
"apricotdog"
"bananacat"
"lycheecat"
"apricotcat"
"bananamouse"
"lycheemouse"
"apricotmouse"
"bananaelephant"
"lycheeelephant"
"apricotelephant"
"bananacow"
"lycheecow"
"apricotcow"
"bananapig"
"lycheepig"
"apricotpig"
"bananarabbit"
"lycheerabbit"
"apricotrabbit"
"bananalion"
"lycheelion"
"apricotlion"
列量
"banana"+行量
"lychee"+行量
"apricot"+行量
行量
"tiger"
"dog"
"cat"
"mouse"
"elephant"
"cow"
"pig"
"rabbit"
"lion"

【図９】

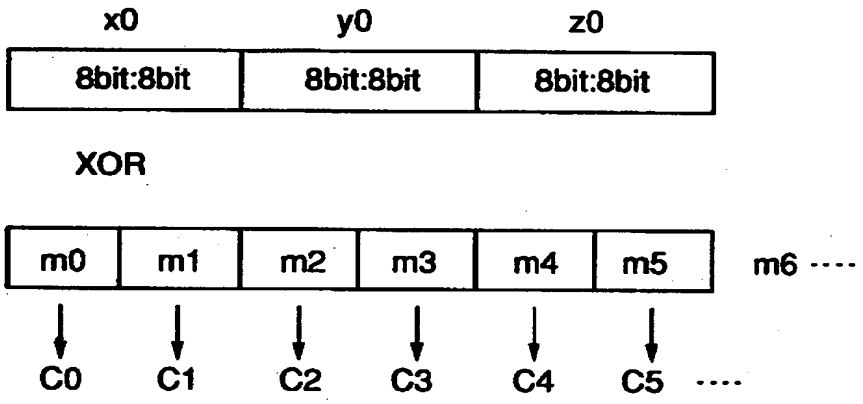


【図１０】



出証特２０００－３０２４１５６

【図１１】


x0
y0
z0
8bit:8bit
8bit:8bit
8bit:8bit
XOR
m0
m1
m2
m3
m4
m5
m6 ----
C0
C1
C2
C3
C4
C5 ----

【書類名】　　要約書

【要約】

【課題】データベースのセキュリティを確保すると共に、検索に利用される項目については高速検索を可能とする。

【解決手段】データベースを暗号化する際に、検索に利用される列項目のデータについては鍵記憶装置１５に記憶された当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては鍵記憶装置１５に記憶された各行毎に固有の行鍵を用いて暗号化してデータ記憶装置１６のデータベース格納エリア１６ａに保存しておく。このように、各行毎に鍵を異ならせることでセキュリティを高めることができ、検索時には、検索用として入力されたデータを前記所定の列項目に共通の列鍵を用いて暗号化し、その暗号化された検索用データとデータベース格納エリア１６ａ内の暗号化データベースとを比較することで高速検索を実現できる。

【選択図】　　図１

出　願　人　履　歴　情　報

識別番号　　〔０００００１４４３〕

１．変更年月日　　１９９８年　１月　９日
［変更理由］　住所変更
住　所　東京都渋谷区本町１丁目６番２号
氏　名　カシオ計算機株式会社